Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S67	33	annotation$2 same memory adj location Rev all	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 17:12
S66	21	annotation$2 same memory adj locations	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 17:12
S65	48	compil$5 same annotation$2 same memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 17:09
S64	7	compil$5 same annotation$2 same (memory near2 location$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 16:03
S63	45	compil$5 same annotation$2 same (architecture$2 or convention$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 15:29
S61	18	compil$5 and annotation$2 same (architecture$2 and convention$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/06/08 15:29
S59	1	compil$5 and annotation$2 same non$ambiguous same (memory or location$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/27 15:41
S55	5	compil$5 and annotation$2 and non$ambiguous	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/27 15:39
S54	8	annotation$2 and non$ambiguous	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/27 15:37

Terms used compiler annotation architecture memory Found 36,696 of 156,259

Sort results by relevance Save results to a Binder Try an Advanced Search

Display results expanded form Search Tips Try this search in The ACM Guide

Open results in a new window

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Session S6.2: compilers and program analysis: PACT HDL: a C compiler targeting ASICs and FPGAs with power and performance optimizations

Alex Jones, Debabrata Bagchi, Satrajit Pal, Xiaoyong Tang, Alok Choudhary, Prith Banerjee

October 2002 **Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems**

Full text available: pdf(340.93 KB) Additional information: full citation, abstract, references, index terms

Chip fabrication technology continues to plunge deeper into sub-micron levels requiring hardware designers to utilize ever-increasing amounts of logic and shorten design time. Toward that end, high-level languages such as C/C++ are becoming popular for hardware description and synthesis in order to more quickly leverage complex algorithms. Similarly, as logic density increases due to technology, power dissipation becomes a progressively more important metric of hardware design. PACT HDL, a C to ...

Keywords: ASIC, FPGA, FSM, HDL, IP, SoC, VHDL, Verilog, compiler, high-performance, levelization, low-power, pipelining, synthesis

2 Performance counters and state sharing annotations: a unified approach to thread locality

Boris Weissman

October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33 , 32 Issue 11 , 5

Full text available: pdf(1.76 MB) Additional information: full citation, abstract, references, citings, index terms

This paper describes a combined approach for improving thread locality that uses the bardware performance monitors of modem processors and program-centric code annotations to guide thread scheduling on SMPs. The approach relies on a shared state cache model to compute expected thread footprints in the cache on-line. The accuracy of the model has been analyzed by simmations involving a set of parallel applications. We demonstrate how the cache model can be used to implement several practical loca ...

3 Track 5: supercomputing (part 2): A case for a working-set-based memory hierarchy

Steve Carr, Soner Önder

May 2005 **Proceedings of the 2nd conference on Computing frontiers**

Full text available: pdf(163.80 KB) Additional information: full citation, abstract, references, index terms

Modern microprocessor designs continue to obtain impressive performance gains through increasing clock rates and advances in the parallelism obtained via micro-architecture design. Unfortunately, corresponding improvements in memory design technology have not been realized, resulting in latencies of over 100 cycles between processors and main memory. This ever-increasing gap in speed has pushed the current memory-hierarchy approach to its limit.Traditional approaches to memory-hierarchy manageme ...

Keywords: cache design, loop tiling

4 An optimal memory allocation scheme for scratch-pad-based embedded systems

Oren Avissar, Rajeev Barua, Dave Stewart

November 2002 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 1 Issue 1

Full text available: pdf(396.62 KB) Additional information: full citation, abstract, references, citings, index terms

This article presents a technique for the efficient compiler management of software-exposed

heterogeneous memory. In many lower-end embedded chips, often used in microcontrollers and DSP processors, heterogeneous memory units such as scratch-pad SRAM, internal DRAM, external DRAM, and ROM are visible directly to the software, without automatic management by a hardware caching mechanism. Instead, the memory units are mapped to different portions of the address space. Caches are avoided due to the ...

Keywords: Memory, allocation, embedded, heterogeneous, storage

5 Cool-Cache: A compiler-enabled energy efficient data caching framework for embedded/multimedia processors

Osman S. Unsal, Raksit Ashok, Israel Koren, C. Mani Krishna, Csaba Andras Moritz

August 2003 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 2 Issue 3

Full text available: pdf(834.37 KB) Additional information: full citation, abstract, references, index terms

The unique characteristics of multimedia/embedded applications dictate media-sensitive architectural and compiler approaches to reduce the power consumption of the data cache. Our goal is exploring energy savings for embedded/multimedia workloads without sacrificing performance. Here, we present two complementary media-sensitive energy-saving techniques that leverage static information. While our first technique is applicable to existing architectures, in our second technique we adopt a more rad ...

Keywords: Low-power design, cache partitioning, compiler-architecture interaction, tagless caching

6 Multimedia and graphics: Cool-cache for hot multimedia

Osman S. Unsal, Raksit Ashok, Israel Koren, C. Mani Krishna, Csaba Andras Moritz

December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(970.61 KB) Publisher Site Additional information: full citation, abstract, references, citings

We claim that the unique characteristics of multimedia applications dictate media-sensitive architectural and compiler approaches to reduce the power consumption of the data cache. Our motivation is exploring energy savings for real-time multimedia workloads without sacrificing performance. Here, we present two complementary media-sensitive energy-saving techniques that leverage static information. While our first technique is applicable to existing architectures, in our second technique we adop ...

7 The benefits and costs of DyC's run-time optimizations

Brian Grant, Markus Mock, Matthai Philipose, Craig Chambers, Susan J. Eggers

September 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 5

Full text available: pdf(1.59 MB) Additional information: full citation, abstract, references, citings, index terms

DyC selectively dynamically compiles programs during their execution, utilizing the run-time-computed values of variables and data structures to apply optimizations that are based on partial evaluation. The dynamic optimizations are preplanned at static compile time in order to reduce their run-time cost; we call this staging. DyC's staged optimizations include (1) an advanced binding-time analysis that supports polyvariant specialization (enabling both single-way and multi ...

Keywords: dynamic compilation, specialization

8 Coupling compiler-enabled and conventional memory accessing for energy efficiency

Raksit Ashok, Saurabh Chheda, Csaba Andras Moritz

May 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 2

Full text available: pdf(1.41 MB) Additional information: full citation, abstract, references, index terms

This article presents Cool-Mem, a family of memory system architectures that integrate conventional memory system mechanisms, energy-aware address translation, and compiler-enabled cache disambiguation techniques, to reduce energy consumption in general-purpose architectures. The solutions provided in this article leverage on interlayer tradeoffs between architecture, compiler, and operating system layers. Cool-Mem achieves power reduction by statically matching memory operations with energy-eff ...

Keywords: Energy efficiency, translation buffers, virtually addressed caches

9 Compiler transformations for high-performance computing

David F. Bacon, Susan L. Graham, Oliver J. Sharp

December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Full text available: pdf(6.32 MB) Additional information: full citation, abstract, references, citings, index terms, review

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

[10] CRL: high-performance all-software distributed shared memory
K. L. Johnson, M. F. Kaashoek, D. A. Wallach
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles**, Volume 29 Issue 5
Full text available: pdf(2.02 MB) Additional information: full citation, references, citings, index terms

[11] Retargetable estimation scheme for DSP architecture selection
Naji Ghazal, Richard Newton, Jan Rabaey
January 2000 **Proceedings of the 2000 conference on Asia South Pacific design automation**
Full text available: pdf(97.60 KB) Additional information: full citation, references, citings

[12] Programming data parallel algorithms on distributed memory using Kali
Charles Koelbel, Piyush Mehrotra
June 1991 **Proceedings of the 5th international conference on Supercomputing**
Full text available: pdf(937.29 KB) Additional information: full citation, references, citings, index terms

[13] High performance Fortran language specification
CORPORATE Rice University
December 1993 **ACM SIGPLAN Fortran Forum**, Volume 12 Issue 4
Full text available: pdf(5.69 MB) Additional information: full citation, abstract, citings, index terms

(PART I)Fortran Forum is reprinting this High Performance Fortran Language Specification over several issues. The current issue is devoted to the first four chapters of the HPFF Language Specification. Remaining chapters of the HPFF Language Specification, and the HPFF Journal of Development, will be printed in installments in filture issues of Fortran Forum.

[14] Comparing data forwarding and prefetching for communication-induced misses in shared-memory MPs
David Koufaty, Josep Torrellas
July 1998 **Proceedings of the 12th international conference on Supercomputing**
Full text available: pdf(1.12 MB) Additional information: full citation, references, citings, index terms

[15] Influence of Memory Hierarchies on Predictability for Time Constrained Embedded Software
Lars Wehmeyer, Peter Marwedel
March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 1**
Full text available: pdf(813.06 KB) Additional information: full citation, abstract

Safety-critical embedded systems having to meet real-time constraints are expected to be highly predictable in order to guarantee at design time that certain timing deadlines will always be met.This requirement usually prevents designers from utilizing caches due to their highly dynamic, thus hardly predictable behavior.The integration of scratchpad memories represents an alternative approach which allows the system to benefit from a performance gain comparable to that of caches while at the sam ...

[16] An evaluation of staged run-time optimizations in DyC
Brian Grant, Matthai Philipose, Markus Mock, Craig Chambers, Susan J. Eggers
May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation**, Volume 34 Issue 5
Full text available: pdf(1.54 MB) Additional information: full citation, abstract, references, citings, index terms

Previous selective dynamic compilation systems have demonstrated that dynamic compilation can achieve performance improvements at low cost on small kernels, but they have had difficulty scaling to larger programs. To overcome this limitation, we developed DyC, a selective dynamic compilation

system that includes more sophisticated and flexible analyses and transformations. DyC is able to achieve good performance improvements on programs that are much larger and more complex than the kernels. We ...

17 Exploiting dual data-memory banks in digital signal processors
Mazen A. R. Saghir, Paul Chow, Corinna G. Lee
September 1996 **Proceedings of the seventh international conference on Architectural support for programming languages and operating systems**, Volume 31 , 30 Issue 9 , 5
Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms

Over the past decade, digital signal processors (DSPs) have emerged as the processors of choice for implementing embedded applications in high-volume consumer products. Through their use of specialized hardware features and small chip areas, DSPs provide the high performance necessary for embedded applications at the low costs demanded by the high-volume consumer market. One feature commonly found in DSPs is the use of dual data-memory banks to double the memory system's bandwidth. When coupled ...

18 Java annotation-aware just-in-time (AJIT) complilation system
Ana Azevedo, Alex Nicolau, Joe Hummel
June 1999 **Proceedings of the ACM 1999 conference on Java Grande**
Full text available: pdf(1.26 MB) Additional Information: full citation, references, citings, index terms

19 Using an architectural knowledge base to generate code for parallel computers
Anthony E. Terrano, Stanley M. Dunn, Joseph E. Peters
September 1989 **Communications of the ACM**, Volume 32 Issue 9
Full text available: pdf(1.01 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The authors present a reconfigurable compiler for distributed memory parallel computers that performs automatic program partitioning, mapping, and communication code generation under the guidance of directives supplied by the programmer.

20 Bidwidth analysis with application to silicon compilation
Mark Stephenson, Jonathan Babb, Saman Amarasinghe
May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5
Full text available: pdf(930.97 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper introduces Bitwise, a compiler that minimizes the bitwidth the number of bits used to represent each operand for both integers and pointers in a program. By propagating 70 static information both forward and backward in the program dataflow graph, Bitwise frees the programmer from declaring bitwidth invariants in cases where the compiler can determine bitwidths automatically. Because loop instructions comprise the bulk of dynamically executed instructions, Bitw ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next